# Mr. GNU's Who

David I. Lewin

Citation: Computers in Physics **11**, 406 (1997); doi: 10.1063/1.4822578
View online: https://doi.org/10.1063/1.4822578
View Table of Contents: https://aip.scitation.org/toc/cip/11/5
Published by the American Institute of Physics

---

**ARTICLES YOU MAY BE INTERESTED IN**

Calculus on the Web
Computers in Physics **11**, 415 (1997); https://doi.org/10.1063/1.4822582

Online Journal Section Gets There First
Computers in Physics **11**, 415 (1997); https://doi.org/10.1063/1.4822581

“Laser Adventure”: Converting Print Material to a Web Site
Computers in Physics **11**, 411 (1997); https://doi.org/10.1063/1.4822580

Developing A Computer-Rich Physics Curriculum at a Liberal-Arts College
Computers in Physics **11**, 436 (1997); https://doi.org/10.1063/1.4822586

Catching the Right Bus, Part VII: A Camac Example
Computers in Physics **11**, 425 (1997); https://doi.org/10.1063/1.4822584

SC97 to Examine Interdisciplinary Applications
Computers in Physics **11**, 408 (1997); https://doi.org/10.1063/1.4822579

---

AIP Conference Proceedings
FLASH WINTER SALE!
50% OFF ALL PRINT PROCEEDINGS
ENTER CODE 50DEC19 AT CHECKOUT

## An Interview with

# Richard Stallman

## Mr. GNU's Who

In the beginning, software was born free—as the Swiss/French philosopher Jean-Jacques Rousseau might have put it—from the minds of a community of programmers. But today, according to Richard Stallman, it is everywhere in chains—the chains of copyright and patent law. Most software is now proprietary, distributed only as compiled code that cannot be legally altered and shared by users.

Stallman—the man who created Emacs, the Swiss army knife of Unix text editors—wants to restore to programmers and users of software alike the freedom to tinker that originally attracted many brilliant programmers to the field. As leader—Chief GNUisance—of the GNU Project, a volunteer effort to develop a nonproprietary operating system, he wants all software users to have the right to share the experience of taking someone else's code, analyzing it, and adding their own improvements. In doing so, they may come to experience the same freedom and sense of community that he experienced as a systems programmer at the Artificial Intelligence Laboratory (AI Lab) at the Massachusetts Institute of Technology (MIT) in the 1970s and early 1980s. GNU (pronounced with a hard "G"), he says, stands for "GNU's not Unix."

"I'm not opposed to copyright" being applied to software, says Stallman. "I'm opposed to nonfree software. Does it make sense to motivate people to do work through a mechanism that restricts this work?" Stallman also believes that struggles between corporate interests and the public over the expansion of intellectual property to include software, genetic information, and personal data will grow. "This is one of the major factors in concentrating wealth" globally, he says, "and it will be one the major conflicts of the next century."

Through the GNU Project, Stallman has created a worldwide empire of "free software"—software distributed not necessarily gratis, but with its source code so that is open for modification by anyone, and also open for redistribution by anyone willing to provide the modified version together with *its* source code. To spread the concept of nonproprietary (free) software, Stallman founded and became president of the Free Software Foundation in Cambridge, MA.

How successful has GNU been? The kernel of the operating system—GNU's HURD (for Herd of Unix Replacement Daemons)—is still a work-in-progress, in its third test release as part of a runnable GNU system. However, other components of the operating system such as its compilers, C++ library, and shell are already in widespread use. In addition, the thousands of people who are using Linux, the operating system named for programmer Linus Torvalds, are actually using a version of GNU without knowing it, Stallman says. Torvalds's contribution to Linux was to write an alternative kernel for GNU. The GNU project was already under way, but its ambitious plans to create an enhanced form of Unix required more time than Torvalds's one-person effort to write a free Unix kernel. Subsequently, Linux was combined with the rest of the GNU system to form a complete operating system.

The operating system is only the initial software needed to make a computer usable, Stallman notes. "Now that we have a free operating system, we want to have free everything else," he says. The foundation is developing a variety of applications, including a formerly non-free drawing program (Leonardo), a spreadsheet, and a GNU SQL server, and it plans to develop a GNU Web browser. By providing a basic set of free applications, Stallman hopes to spread the idea that users of all software are entitled to the freedoms that are embodied in the GNU General Public License.

Critics have asked how programmers could support themselves in a world where software was free. Stallman notes that there are companies today that repackage and sell free software on CD-ROM disks, or that make money by providing support for users. In fact, much of FSF's income comes from the sale of software and manuals—even though the software is freely available at the foundation's Web site and in several Internet software archives worldwide.

Stallman became interested in computers early, while at summer camp in the 1960s. "As soon as I heard of the idea of computers I was fascinated by them," he says. At first, he only had access to computer manuals that some of the counselors had brought with them to the camp—but not to the machines themselves. Later, in high school, he gained access to a mainframe at the IBM New York Scientific Center in Manhattan.

"I was fascinated with compilers," Stallman says. He began writing a preprocessor—a program whose input is another program and whose output is sent to the compiler—for a PL/I compiler, to add a feature to the compiler. "Perhaps this was a silly idea, since PL/I had too many features to start with," he says. As he learned more about programming, he lost interest in the project. But he was noticed by the staff at the IBM center, who offered him a summer job.

Subsequently, Stallman left New York for Cambridge, MA, to enroll at Harvard as an undergraduate. At that time, the computing program at Harvard "wasn't very good," he says. So he traveled around the Boston area dropping in on any site with a computer, at one point visiting the AI Lab to study the manuals for its computers. "They didn't have much in the way of manuals, so they gave me a job instead," Stallman says. There, he had the opportunity to modify the operating system then in use, the Incompatible Time-sharing System (ITS) that

© 1997 AMERICAN INSTITUTE OF PHYSICS S0894-1866(97)01205-4

had been developed at MIT. "That was a really exciting thing, much more exciting than my course work," he says.

Eventually—after graduating from Harvard in physics—he joined the programming staff of the AI Lab as a permanent employee. In many ways, including his e-mail address (rms@gnu.ai.mit.edu), Stallman has never left.

The AI Lab had a very different spirit from that of most academic laboratories. In contrast to the "extremely hierarchical and bureaucratic atmosphere" for computing that Stallman experienced at Harvard, resources at the AI Lab were available to all as needed. This atmosphere was maintained by the hackers and encouraged by the lab's director, artificial-intelligence pioneer Marvin Minsky. On more than one occasion a faculty member locked an office, including a terminal, perhaps to protect some papers. The system hackers needed the terminal and broke into the office to gain access. Thereafter, the professors secured what they needed to, but left their offices open, Stallman says.

"Being a hacker at the AI Lab meant having an attachment to certain ideas of freedom and of how to treat other people," Stallman says. You could walk into a room with a stereo and put on some music, walk into a room with a computer terminal and log in, or just sit down and read a book, reminisces Stallman, to whom the lab felt like his home.

In general, the hackers worked on maintaining and upgrading the operating system, only occasionally writing code for the AI or robotics projects under way at the lab. "It wasn't for theoretical research; we were developing a system to be used," Stallman says. "We added features [to ITS] because we wanted to use them"—although the programmers did pay attention to the expressed needs of nonhacker users of the system, he acknowledges.

By the early 1980s the atmosphere changed as Symbolics Inc., one of two competing MIT spinoff companies developing computer workstations for artificial intelligence, recruited many of the systems hackers from the AI Lab, Stallman says. "With most of the hackers gone, that culture was no longer viable," he notes, "and the mainstream culture that graduate students and professors had brought from other places swept though the lab." At the same time, MIT purchased new mainframes from Digital Equipment Corp. that used DEC's proprietary operating system. "All of our software became obsolete at a time when the people who could port it to the new computer were no longer around," he says.

Stallman stayed on for two more years, to keep the AI Lab from becoming totally dependent on Symbolics for operating software for its AI workstations. Having succeeded in hiring many of the systems hackers, Symbolics was pressuring the AI Lab to support it in its competition with the other startup, Lisp Machines Inc. "I was outraged at the violation of my neutrality," says Stallman. But working alone for two years to maintain the workstation software became untenable. Stallman's resistance, however, gave Lisp Machines a fair shot at the market.

"I wanted to recreate what was lost"—the hacker ethic—"but that had to be done in a way that spread it beyond a single site," Stallman says. The answer was to create a hacker community without geographic bounds. And to do that, the hackers would need to have a nonproprietary operating system. "An operating system was the obvious choice to work on first," he says. This led him to quit his job in early 1984 to begin work on GNU—and in 1985 to the establishment of the Free Software Foundation.

"I wasn't sure when I quit my job how I would support myself," Stallman says. "But as soon as I had free software that people wanted to use, ways of raising money appeared." One source of funds was to sell the software on tape; another was teaching classes on how to use Emacs and providing other forms of support. He eagerly received donations of equipment—and even of a year's salary for a programmer, which he used to hire someone else. "That's when I made a decision that I've stuck by: The Free Software Foundation doesn't pay me," Stallman says.

Then in 1990, Stallman was named a MacArthur fellow, receiving a five-year "genius" grant that gave him—and the free-software cause—great visibility in the media. "It caused some people to treat me with more respect," Stallman says, and allowed him to focus his efforts on FSF and the GNU project, rather than on making a living.

Even now, Stallman lives in large part on the interest from that award. He continues to live cheaply, which allows him to do what he feels is important without compromising his principles. "A lot of the luxuries that people think they need strike me as traps," he says. "In an ironic way, by working on free software, I have become in a sense independently wealthy—as long as I live frugally."

**David I. Lewin**